WEP
Warwick Evans Publishing

Transactions on Environment, Energy and Earth Sciences
ISSN: 2960-1924, eISSN: 2960-2300 | Volume 5
EMCEME 2025

Improving the Practical Experience of VR Games Based on Custom Gesture Interaction—Taking Minecraft as an Example

Jinrun Wan *

International School of Information Science & Engineering, Dalian University of Technology, Dalian, Liaoning, 116600, China

* Corresponding Author Email: *wdnmdnmsl@mail.dlut.edu.cn*

Abstract. Although virtual reality (VR) games have developed rapidly, they still face the problem of low user retention rates due to poor interactive experiences. To address the issues of VR users' immersive experience and the operational efficiency of VR games, this study takes the VR mod *Vivecraft* of *Minecraft* as an example and proposes a user-defined gesture system that integrates a shortcut key operation mode and a gesture mapping framework, and demonstrates the feasibility of this solution through the classic operations of "flying in the game" and "reloading in the game" as cases. The research results show that this solution can significantly enhance players' sense of immersion and the operational feel within the game. However, the study also finds that user-defined gestures have problems such as gesture misjudgment, insufficient gesture accuracy, and high development difficulty, which need to be solved with multiple references in the future. This study provides theoretical and practical references for VR gesture interaction design.

Keywords: VR gesture interaction design; Shortcut key; Gesture customization; Minecraft.

1. Introduction

In recent years, virtual reality (VR) technology has continued to develop in the gaming field, with continuous advancements in hardware iteration and content innovation. However, the market acceptance of VR gaming models remains insufficient. According to the VR headset usage survey report released by Steam in July 2025, only 1. 41% of active users in June 2025 used VR devices. The reasons for such a low usage rate mainly lie in two aspects:

In terms of software, many mainstream game types cannot be fully adapted to the VR version. For example, MOBA games, which mainly offer a bird's-eye view of the battle situation, cannot be changed to the first-person perspective, and high-freedom gestures cannot be used for MOBA’s abilities. In terms of hardware, existing devices generally have the defects of inducing dizziness and visual fatigue, causing most users to still prefer keyboard and mouse, or gamepad [1].

In response to the above problems, many game developers have attempted to enhance the in-game immersion and player experience to improve player retention. In the VR piano playing teaching game, Min Zeng used the Hidden Markov Models algorithm and gesture recognition technology to identify players' actions of imitating piano playing and correct problems, thus improving teaching effectiveness [2]. When developing the guqin VR game, Yuting Huang considered both the difficulty of getting started with the guqin and the game's realism, adopting a differentiated design. She redesigned four basic guqin gestures--"tiao", "gou", "ring-finger stop", and "thumb stop"-as the judgment gestures in the VR game [3]. In the design process of the VR rhythm game, Riski Nur Azizah et al. used Kinect to record dancers' action skeletons and generate a database. After processing, these data can serve as the judgment conditions for in-game note generation [4].

Although the above-mentioned games can trigger relevant reactions through postures or gestures in their specific modes, the game will not respond as long as the gestures or actions deviate from the pre-determined ones in the game. For most VR games, they do not have a very complex system, and pre-set actions can meet the game requirements. However, the sandbox games require a large amount of interaction, and pre-set actions cannot meet players' needs, and solutions outside the pre-set ones also cannot achieve a fully immersive gaming experience.

In response to the above problems, this study takes Minecraft as an example, combining "customized shortcut key usage" with "VR gesture interaction", which can not only meet users' personalized needs but also enhance the immersion of VR games.

2. Overview of related technologies

2.1. Theoretical basis of shortcut keys

2.1.1. Overview of Shortcut Keys

Shortcut keys, also known as hot keys, are an interactive way to execute specific commands through single-key or multi-key combinations on the keyboard. Compared with mouse operations, shortcut keys can perform complex instructions with a more concise input path. Users only need to press the preset key combinations (such as PrtSc for taking screenshots and Ctrl+C for copying) to directly trigger the corresponding functions, without having to complete the operations through multi-level menus or precise clicks [5].

2.1.2. Combination commands of modifier keys and function keys

Each key on the keyboard plays an important role respectively. Among them, the modifier keys such as Ctrl, Alt, and Shift are used in conjunction with other function keys, enabling some keys that originally perform letter input (such as the x, c, and v keys) to serve other functions [6].

2.1.3. Customize shortcut keys

In Windows systems and all large-scale application software, shortcut keys are basically automatically set by default for many of their commands. However, different industries have their own characteristics, different software also has different commands and settings, and different people have different operating habits. Taking the spreadsheet in the software WPS as an example, we can set Alt + Shift + F to achieve the effect of "filling the table color" [5].

2.2. Theoretical Foundation of Gesture Interaction

Gesture interaction, as an important paradigm of human-computer interaction, triggers device responses by recognizing users' gestures that mimic real-life postures, enabling natural and intuitive non-verbal control. In VR games, according to the contact mechanism between the hand and virtual objects, the interaction forms are shown in Table 1 [3]:

Table 1. Explanation of interaction forms.

Interaction form	Trigger condition	Typical applications
Collision-based interaction	The user's hand physically collides with the virtual object.	Object grasping (e. g. , holding a weapon) Spatial displacement (e. g. , dragging building modules) Shape change (e. g. , extrusion deformation)
Pose-based interaction	Users use specific gestures and movements.	Interface operations (e. g. , clicking on the menu / pinching to zoom) Environmental control (e. g. , waving to switch scenes) Symbolic instructions (e. g. , drawing a circle to activate skills)

Its design principle emphasizes consistency with real-world interaction habits. For example, mapping the "pinch" gesture to the zoom function conforms to users' experience transfer of touch-screen operations [7].

In this study, we focus on pose-based interaction and propose a gesture-customizable interaction mode to address the existing problems in the *Minecraft* VR mod Vivecraft, aiming to enhance the naturalness of user interaction and operational efficiency in the virtual environment.

2.3. The current technological status of Vivecraft

Vivecraft is a mod that transforms Minecraft into a special VR experience for room-scale or seated gameplay. It changes almost all the execution methods in the game. From mining minerals to fighting enemies, there are almost corresponding actions in VR. Relevant examples are shown in Table 2:

Table 2. Examples of gesture adaptation of VR for each operation.

Game media	Mine blocks	Slash at the enemy	shoot an arrow
Computer execution mode	When holding the pickaxe, align the crosshair with the corresponding block and press and hold the left mouse button.	When holding the sword, aim the crosshair at the enemy target and click the left mouse button.	When holding the bow, press and hold the right mouse button to charge. Release the mouse, and the arrow will fly out.
VR execution mode	When holding the pickaxe, align the crosshair with the corresponding block and swing the controller on the side of your dominant hand up and down.	When holding the sword, aim the crosshair at the enemy target and continuously swing the controller on the side of your dominant hand.	While holding the bow, long-press the interactive button on the handle. First, place both hands in front of your chest. Then, stretch your left hand forward and move your right hand backward, resembling the shape of drawing a bow.

Although Vivecraft is compatible with almost all the content that should be in the original version, there are obvious problems in its compatibility with other mods. Facing the many specific shortcut keys in mods (such as pressing M to open the map in the Journey Map mod), VR controllers cannot correspond to these keys one by one. The solution provided by Vivecraft for this problem is also not satisfactory: players must bind the operations corresponding to each section of the radial menu in the settings and then open the radial menu in the game to select the corresponding operation, as shown in Figure 1. This approach is not conducive to enhancing the player's immersion in the game. Therefore, this study proposes a solution of custom gesture interaction for such problems of Vivecraft.

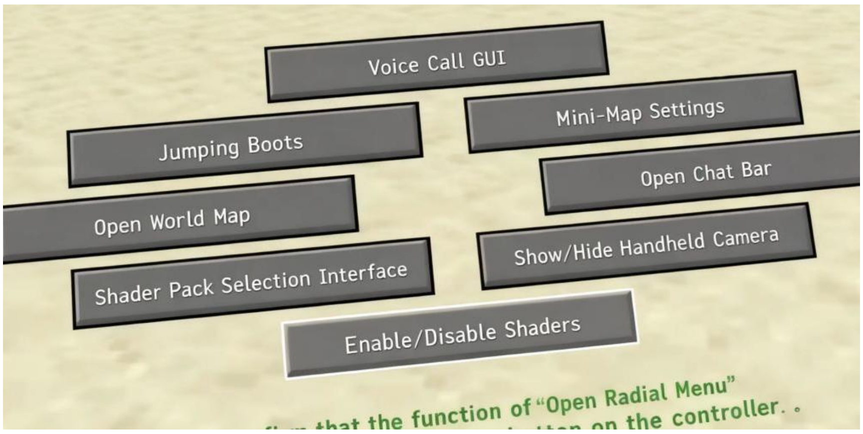

Figure 1. Radial menu of Vivecraft.

3. Custom gesture system design

This research is based on the VR mod Vivecraft of *Minecraft* to design and implement a user-customizable gesture interaction system. This system allows players to map common game operations to gestures according to their own operating habits and task requirements, thereby improving operation efficiency and immersion. The system architecture is divided into a gesture recording module, a mapping management module, and a real-time recognition module, supporting players to freely define, modify, and test gesture actions in the virtual environment.

In the specific implementation process, this research selects two typical operation scenarios in *Minecraft* for case design, namely "original basic operations" and "mod expansion operations", to verify the applicability of the system in the native gameplay and community mods.

3.1. Original basic operations

Although Vivecraft has rewritten every operation in the original version, some operations lack intuitiveness and immersion. This study needs to perform compatible operations on this. Even for the existing operation system of Vivecraft, players can still change the actions corresponding to the operations to achieve the "immersion" they desire.

3.1.1. Customization of flight gestures

In the original creative mode, players need to double-click the spacebar to take off and hover. In VR, players need to double-click specific buttons on the controller. The in-game flight mode is shown in Figure 2. Although players cannot take off in reality, the "double-click" operation is less intuitive for taking off, prone to accidental touches, and can disrupt the immersive experience.

This study designs a "pushing-up" motion as the take-off gesture: players can synchronously push their hands upwards as the flight activation gesture, as shown in Figure 3. This action draws on the "lifting" metaphor in reality and conforms to operating habits, that is, users tend to interact using actions consistent with real-world physical experiences [7].

Figure 2. Schematic diagram of in-game flight.

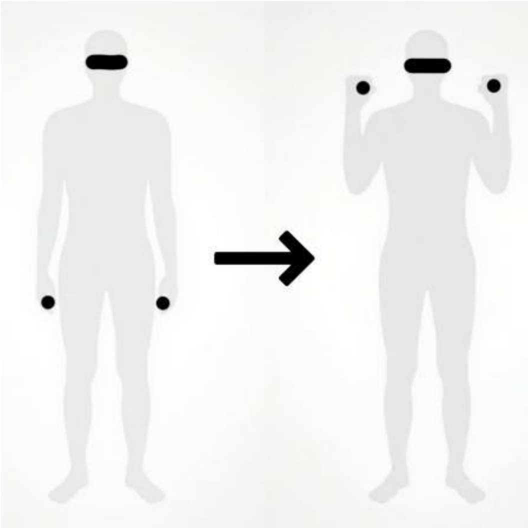

Figure 3. Schematic diagram of customizing flight gesture.

3.2. Mod Extension Operations

As mentioned earlier, Vivecraft does not fully support all mods. For many operations in mods, players need to open the radial menu to make selections. This operation not only fails to highlight the differences between VR games and the original game but may even make players feel that "it's not as good as the original version". To address this issue, this study integrates the gesture customization system into various mods. This not only allows players to create their most suitable gestures but also significantly enhances the freedom of sandbox games and the immersive experience for players.

3.2.1. Customization of Reload Gestures (Taking MrCrayfish's Gun Mod as an Example)

In the original mod, the reload operation requires pressing the R key on the keyboard. In VR, it is realized by opening the circular menu and selecting the reload option. This mode lacks feedback and realism. The performance of MrCrayfish's Gun Mod in Vivecraft is shown in Figure 4.

This study designs a "gun-swinging" action as the reload gesture: the player quickly swings the gun-holding hand downward, simulating the action of swinging the gun to eject the cartridge case in reality, as shown in Figure 5. This design emphasizes the behavioral metaphor and personalized expression of the gesture [7].

It is estimated that this gesture can not only significantly improve the realism and smoothness of the operation, but also adapt to different weapon mods through the system mapping management layer, indicating that the system should have good mod compatibility and scalability.

Figure 4. Schematic diagram of the compatibility between Vivecraft and MrCrayfish's Gun.

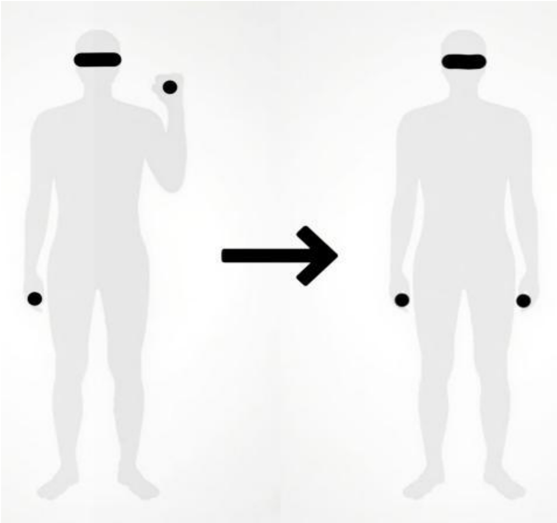

Figure 5. Schematic diagram of the customized reloading gesture.

4. Discussion

The custom gesture interaction system proposed in this study shows significant advantages in improving operation efficiency and immersion. However, it is also necessary to face up to the problems that may arise in its practical application.

4.1. Gesture misjudgment problem

In the diverse and complex environment of sandbox games, the system's discrimination of various gestures and players' misuse of gestures in high-pressure scenarios are quite prominent. The gesture recognition system relies on sensor accuracy, algorithm robustness, and the consistency of user actions. Any deviation in these aspects may lead to misrecognition, which in turn affects the game experience. For example, when a player swings a sword up and down with the right hand, if the player has designed a reload action as shown in Figure 5, the system may misjudge that the player has used the reload operation. In the future, it is necessary to use gesture recognition methods such as Open CV to improve the accuracy and efficiency of gesture recognition by calculating the Euclidean distance between the participant's actions and the standard images of VR games [8].

4.2. Gesture precision problem

In the process of system development, it is necessary to combine the virtual and the real with the help of various devices and software. Among them, Kinect can capture actions without actual physical contact with the input hardware. It can recognize large-scale arm and body movements but cannot record fine hand movements. If a player designs the action of opening the map as "changing the left hand from a fist to an open state", Kinect will not be able to capture the action. In the future, it is necessary to use more precise facility hardware for auxiliary development and more accurate gesture interaction algorithms to optimize the system mode [9].

4.3. Difficulty in gesture system development

Compared with traditional system development, gesture system development requires debugging and coding in the VR environment, and many problems will occur in this process: inconsistent display devices, cognitive differences in the interaction experience, and the irrationality of three-dimensional information coding. These problems not only reduce the development efficiency but also affect the usability, interactivity, and immersion of the developed application programs. In the future, it may be necessary to use the visual programming system of the gesture interaction interface in virtual reality to simplify the development of the gesture system [10].

5. Conclusion

This study takes the VR version of *Minecraft* as an example and designs a set of user-customizable gesture interaction systems. The results show that the system can significantly improve the operation efficiency and sense of immersion, effectively reduce the cognitive load, and successfully transfer the "shortcut key logic" in the two-dimensional interface to the three-dimensional interaction environment, enriching the theoretical and practical methods of VR gesture interaction.

However, the research also shows that the customizable gesture system still faces various problems in practical applications, mainly reflected in misjudgment, accuracy, development difficulty, etc. This problem reveals the inherent tension between personalized interaction and system stability.

Future research should focus on solving problems from both the user's gaming experience and the developer's debugging difficulty by introducing gesture recognition methods, visual programming systems, etc. Customizable gesture interaction will still be an important direction for optimizing the VR game experience, but its maturity and popularization still require the joint promotion of interdisciplinary cooperation and technological iteration.

References

[1] Chen, Q. J. & Zhong, J. M. (2021). VR technology and the development of China's game industry. Dong Yue Tribune, 42, 9, 93-103+2+191-192. https://link.cnki.net/doi/10.15981/j.cnki.dongyueluncong.2021.09.010

[2] Zeng, M. (2024). Gesture Recognition Technology of VR Piano Playing Teaching Game based on Hidden Markov Model. International Arab Journal of Information Technology. 21, 4, 760-772.

[3] Huang, Y. T. (2024). Research on Gesture Interaction Designin VR Game. Jiangnan University. DOI:10. 27169/d. cnki. gwqgu. 2024. 001049.

[4] Azizah, R. N., Lee, Y. S., Jung, J. J., et al. (2021). Gesture recognition for note generation in vr rhythm game. International Conference on Information Networking, IEEE, 2021, 521-524.

[5] Yu, S. Z. (2007). Discussion on the operational habit andshortcut key in computer teaching. Journal of Guangxi University (Natural Science Edition), 32, 360-362+372. https://link.cnki.net/doi/10.13624/j.cnki.issn. 1001-7445.2007.s1.111

[6] Luo, K. L. & Shi, X. M. (2000). About Shortcut Key. Computer learning, 4, 48. https://link. cnki.net/doi/CNKI: SUN:DLXZ.0.2000-04-030

[7] Zhang, S. Y. (2023). Research on VR natural gesture interaction design based on user experience. Shandong University. DOI:10. 27272/d. cnki. gshdu. 2023. 002254.

[8] Liu, S. Y. (2024). Deep Learning-Based Gesture Recognition in VR Games. Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability, IEEE, 2024, 310-313.

[9] Ahmed, N. & Lataifeh, M. (2023). Natural Hand Gesture Interaction for 3D Transformations in Immersive VR. World Conference on Information Systems and Technologies, Cham: Springer Nature Switzerland, 2024, 216-222.

[10] Xiao, W. Y., Duan, J. Z. & Xue, C. Q. (2020). Research on visual programming system of gesture interaction interface in VR. Proceedings of the 2020 4th International Conference on Big Data and Internet of Things, 2020, 38-42.